

## i Front page

UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Examination in:

IN1900 — Introduction to programming with scientific applications

IN1900 — Introduction to programming for chemists

MAT-IN1105 — Programming, Modelling and Computations

INF1100 — Introduction to programming with scientific applications

Day of examination: October 10th 2017

Examination hours: 09.00-13.00

Attachments: None

Permitted aids: None

- Read through the complete exercise set before you start solving the individual exercises. If you miss information in an exercise, you can provide your own reasonable assumptions as long as you explain them in detail.
- Most of the exercises result in short code where there is little need for comments, unless you do something complicated or non-standard. In that case, comments should convey the idea behind the program constructions such that it becomes easy to evaluate the solution.
- Many exercises ask you to “write a function”. A main program calling the function is then not required, unless it is explicitly stated. You may, in these types of exercises, also assume that necessary modules are already imported outside the function. On the other hand, if you are asked to write a complete program, explicit import of modules must be a part of the solution.

- The maximum possible score on this exam is 25 points. There are 14 questions in total, and the number of points is specified for each individual exercise.

**1(a) What is printed?**

What is printed in the terminal when this program is run?

```
a = 4
```

```
b = a
```

```
a = a+b
```

```
print(a)
```

**Fill in your answer here**

Format ▾ | ↺

✎ | Σ | ✕

Words: 0

Maximum marks: 1

**1(b) What is printed?**

What is printed in the terminal when this program is run?

```
A = [[-1,0,1],[0,0,0],[10,9,8]]
print(A[1][-1])
```

**Fill in your answer here**

Format ▾ | ↺ | ✎ | Σ | ✕

Words: 0

Maximum marks: 1

**1(c) What is printed?**

What is printed in the terminal when this program is run?

```
x = 6
```

```
y = -2
```

```
c = x >= 10 or y != -2
```

```
print(c)
```

**Fill in your answer here**

Format ▾ | ↺ | ✎ Σ | ✕

Words: 0

Maximum marks: 1

**1(d) What is printed?**

What is printed in the terminal when this program is run?

```
import numpy as np
a = [1,2,3]
a_np = np.array(a)
print(a*2)
print(a_np*2)
```

**Fill in your answer here**

Format ▾ | ↺ | ✎ Σ | ✕

Words: 0

Maximum marks: 1

**1(e) What is printed?**

What is printed in the terminal when this program is run?

```
S = 0
for i in range(3):
    S += i**2
print(S)
```

**Fill in your answer here**

Format ▾ | ↺ | ✎ Σ | ✕

Words: 0

Maximum marks: 1

**1(f) What is printed?**

What is printed in the terminal when this program is run? If you believe nothing is printed then write this explicitly in your answer, for instance "Nothing is printed".

```
import sys
A = [['-1','0','1'],['0','0','0'],['10','9','8']]
```

```
try:
    b = int(A[2])
except IndexError:
    print('A has length %d' %len(A))
    sys.exit(1)
except TypeError:
    print('Cannot convert %s to int' %A[2])
    sys.exit(1)
print(b)
```

**Fill in your answer here**

Format ▾ | ↺

✎ Σ | ✕

Words: 0

Maximum marks: 1

### 1(g) **What is printed?**

What is printed in the terminal when this program is run? If you believe nothing is printed then write this explicitly in your answer, for instance "Nothing is printed".

```
def poly(x,k):
```





You have a file named 'data.txt' that contains weather data for 25 december at Blindern, in the format listed below. The file starts with data from 1950 and continues to 2008 (the table below does not show the entire file).

```
# Station, year, mean-temp, min-temp, max-temp
18700 1950 -5.9 -8.3 -2.6
18700 1951 6.0 4.5 7.3 0
18700 1952 -1.4 -1.8 -0.2
18700 1953 -0.2 -1.2 6.0 0
18700 1954 -8.5 -9.9 -3.8
18700 1955 -4.0 -7.7 -0.4
18700 1956 -4.9 -5.8 -4.7
18700 1957 -0.4 -1.7 1.9 0
18700 1958 -0.2 -1.0 0.9 7
18700 1959 2.1 0.8 3.5 0
18700 1960 -3.6 -9.7 -2.6
18700 1961 -6.0 -9.0 -2.1
18700 1962 -9.5 -11.3 -7.7
18700 1963 -2.7 -4.1 -0.9
18700 1964 -8.0 -9.5 -6.0
18700 1965 -2.0 -3.5 -1.0
18700 1966 -6.2 -10.6 -1.4
18700 1967 -6.5 -9.3 -5.2
18700 1968 -1.5 -2.8 0.1
```

Write a program that reads data from the file 'data.txt' and makes four lists of data from the columns 'year', 'mean-temp', 'max-temp' and 'min-temp'. Include necessary imports.

## Fill in your answer here

1	
---	--

Maximum marks: 3

## 2(b) Plot data

Extend the program from question 2a, and plot the three temperatures 'mean\_temp', 'min\_temp' and 'max\_temp' as a function of year. The axes of the plot shall be marked with 'Year' and 'Temperature', and there shall be a legend to explain each curve. Include necessary imports.

You may assume that you write the code for the plot in the same file as the code in question 2a, so the four lists are already available.

**Fill in your answer here**

1	
---	--

Maximum marks: 3

### 3(a) Function

Write a python-function  $piecewise(x, a, b)$  that implements the function:

$$f(x) = \begin{cases} 0.0 & \text{for } x \leq a \\ \frac{x-a}{b-a} & \text{for } a < x \leq b \\ 1.0 & \text{for } x > b \end{cases}$$

Here  $x$ ,  $a$ , and  $b$  are scalar values (numbers, not arrays or lists).

**Fill in your answer here**

1

Maximum marks: 3

### 3(b) Test function

Write a test function `test_piecewise()` that tests the function from question 3a. Set  $a = 0$  and  $b = 1$ , and choose three different values for  $x$ ;  $-1.0$ ,  $0.5$ , and  $1.5$ . The result of these three arguments shall be compared with the expected values  $0.0$ ,  $0.5$  and  $1.0$ .

You can assume that the function `piecewise()` is available in the same file as the test function, so you don't have to write it again.

**Fill in your answer here**

1

Maximum marks: 3

### 3(c) User input

Write a program that reads the values  $x$ ,  $a$ , and  $b$  from the command line, calls the function from 3a with these parameters, and prints the result to the screen. Include a try-except block that handles the two cases that there are not enough input arguments, and that the input arguments have the wrong format. In both cases the program shall print an error message and exit. The error message shall be different for the two types of errors. Include necessary imports.

You can assume that the function from 3a is available in the same file as your program, so you don't have to import it or write it again.

**Fill in your answer here**

1	
---	--

Maximum marks: 3

#### 4(a) Compute a sum

Write a Python function `pi_approx(n)`, which implements the sum

$$s_n = 4 \sum_{k=1}^n \frac{(-1)^{k+1}}{2k-1}$$

Write code for calling the function for `n=10` og `n=100` and printing the result to the screen.

**Fill in your answer here**

```
1 |
```

Maximum marks: 3



## 4(b) Plot function values

Write a program that generates a list of n-values from 1 to 50, calls the function from 4a for all the values, and plots the function values as a function of n. Include necessary imports.

You can assume that the function from 4a is available in the same file as your program, so you don't have to import it or write it again.

**Fill in your answer here**

1	
---	--

Maximum marks: 3